

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22

8

Q

9  
10

10

11

12

## **Listing of the Claims**

1. (Currently Amended) A method for debugging a target computer that utilizes virtual memory paging, the method comprising:

replicating on a host computer an address translation table of the target computer, wherein the address translation table establishes on the host computer correspondences between physical memory addresses and virtual memory addresses that existed on the target computer;

transferring physical memory data from the target computer to the [[a]] host computer; and

replicating virtual memory data from the physical memory data on the host computer, wherein the virtual memory data on the host computer is identical to virtual memory data on the target computer.

2. (Original) The method as recited in claim 1, further comprising debugging a fault on the target computer by analyzing replicated data on the host computer.

3. (Original) The method as recited in claim 1, further comprising caching the replicated data in memory on the host computer.

4. (Original) The method as recited in claim 1, wherein the target computer includes an operating system that uses table-driven paged memory management.

1           5.       (Previously Presented)       The method as recited in claim 1,  
2 wherein:

3           the target computer includes a processor that has halted execution; and  
4           the virtual memory data is located in physical memory of the target  
5 computer.

6  
7           6.       (Currently Amended)       A host computing system, comprising:  
8           a processor;  
9           memory;  
10          means for establishing a connection between the memory and memory of a  
11 target computer;  
12          a data retrieval component configured to transfer address table data from  
13 memory of the target computer to the memory;  
14          an address translation component configured to replicate virtual memory  
15 addresses from the address table data in the memory; and  
16          wherein the virtual memory data in the host computing system are identical  
17 to virtual memory data of the target computer for debugging the target computer  
18 when a CPU of the target computer is halted.

19  
20          7.       (Original)       The host computing system as recited in claim 6,  
21 further comprising cache memory configured to store the replicated virtual  
22 memory addresses.  
23  
24  
25

1           8.     (Original)    The host computing system as recited in claim 6,  
2 wherein the host-side address translation component is further configured to  
3 validate the replicated virtual memory addresses.

4  
5           9.     (Original)    The host computing system as recited in claim 6,  
6 further comprising a memory management verifier that verifies that a processor of  
7 the target computing system has memory management enabled.

8  
9           10.    (Original)    The host computing system as recited in claim 6,  
10 wherein the means for establishing a connection between the memory and memory  
11 of a target computer comprises hardware-assisted debug probes.

12  
13           11.    (Currently Amended)    A method, comprising:  
14           accessing address tables from physical memory of a target computer system  
15 by reading the target memory directly through an application program interface;  
16           replicating the address tables on a host computing system; and  
17           using data contained in the address tables to derive virtual address data that  
18 was used on the target computer system, wherein the virtual address data on the  
19 host computer system are identical to virtual address data on the target computer  
20 system.

21  
22           12.    (Original)    The method as recited in claim 11, further comprising  
23 storing the address tables in memory on the host computer system.

1           13.    (Original)    The method as recited in claim 11, further comprising  
2    caching the virtual address data on the host computer system.

3  
4           14.    (Canceled)

5  
6           15.    (Original)    The method as recited in claim 11, further comprising  
7    determining if memory management of a target computer system processor is  
8    enabled.

9  
10          16.    (Original)    The method as recited in claim 11, further comprising  
11   performing the method only if memory management of a target computer system  
12   processor is enabled.

13  
14          17.    (Original)    The method as recited in claim 11, wherein the  
15   accessing further comprises:

16            locating the address tables in physical memory of the target computer  
17   system; and

18            reading the address tables from the target computer.

19  
20          18.    (Original)    The method as recited in claim 11, further comprising  
21   validating the virtual address data to ensure it is identical to the virtual address data  
22   stored on the target computer system.

1           19.    (Original)    The method as recited in claim 11, further comprising  
2 debugging a fault that occurred on the target computer by analyzing the virtual  
3 address data on the host computer system.

4  
5           20.    (Original)    A computer-readable medium containing processor-  
6 executable instructions that, when executed on a processor, perform the method of  
7 claim 11.

8  
9           21-25.       (Canceled)